CORRECT

Group Lab Report 3

Andrea Bruzzone,Thomas Zhang

2016 M02 18

Assignment 1

This task asks for a sampler of twenty opinion polling cities in Sweden without replacement where the probability of a city being chosen is proportional to the population of the city. In a large city the different municipalities (kommuner) of Sweden count as individual cities. We implement this sampler in `R`.

We first import on R just the informations we need and then we compute this function that takes the data set with all the swedish cities and select one using the scheme offered above. The function can be found in the Code part of this report.

The function is then used in this way:

- apply it to the list of all cities
- remove the city selected from the list
- apply the function again to the updated list of cities until we have just 20 cities.

Since the sampling is random, we do not have always the same results, the following list shows an example of cities selected after the procedure described before:

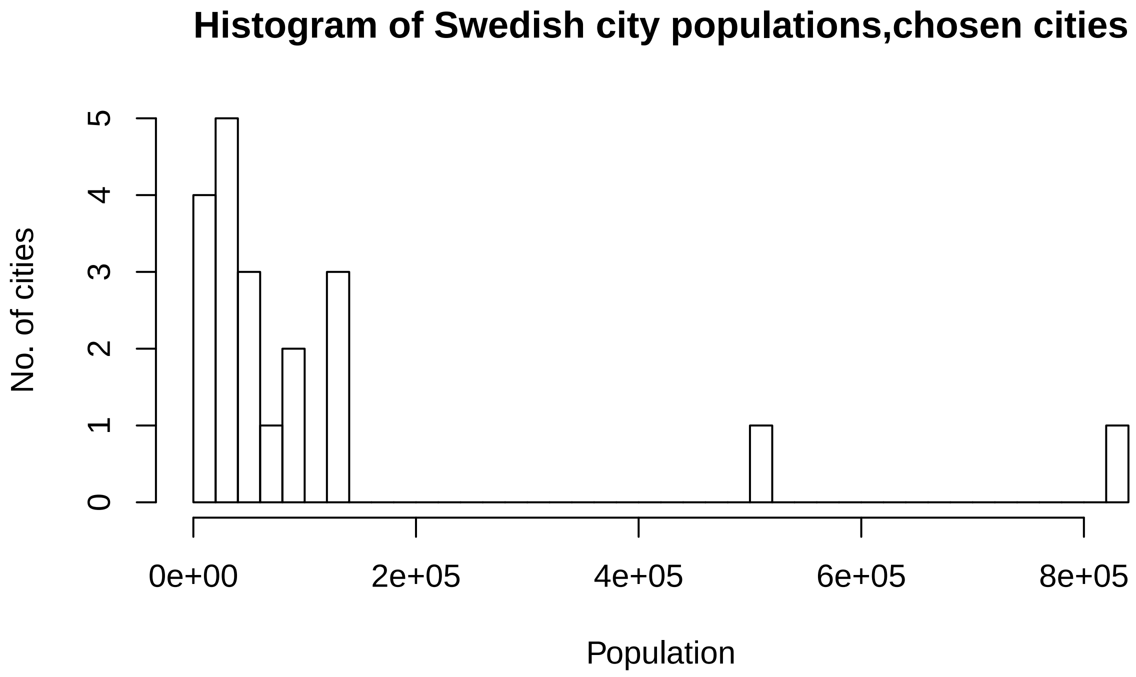
```
##  [1] "Ale"          "Helsingborg" "Piteå"       "Degerfors"   "Örebro"
##  [6] "Strängnäs"    "Stockholm"   "Knivsta"     "Karlstad"    "Göteborg"
## [11] "Borlänge"     "Enköping"    "Sollefteå"   "Norrköping"  "Hässleholm"
## [16] "Skellefteå"   "Simrishamn"  "Sundsvall"   "Värnamo"     "Torsby"
```

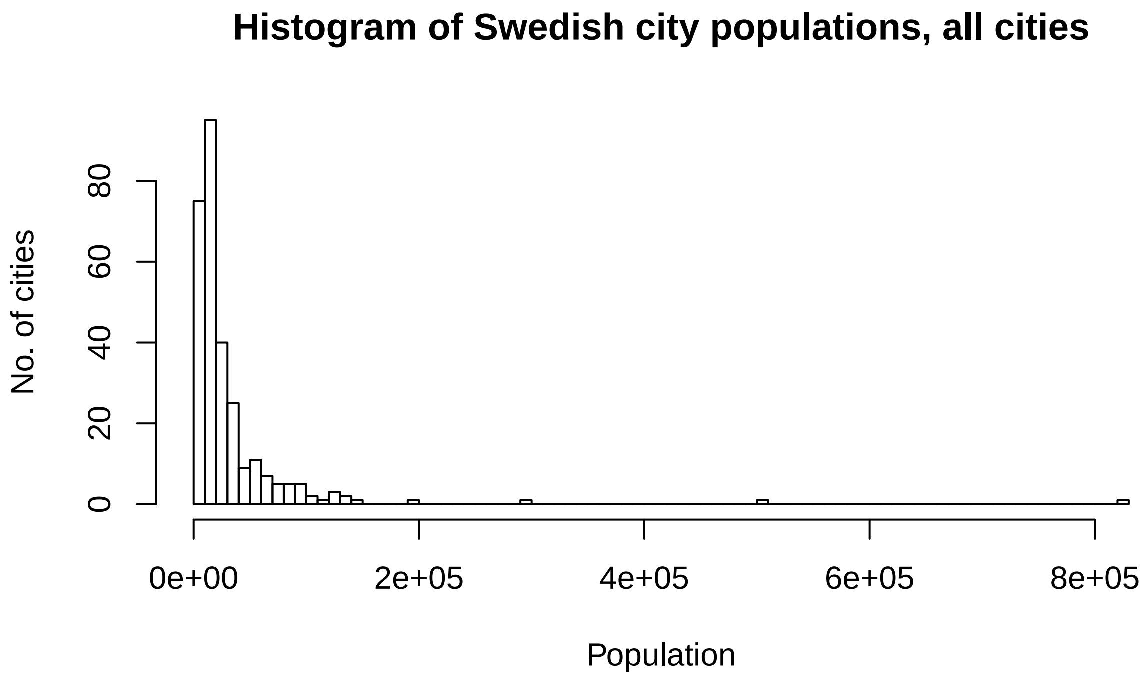
The respective size of the cities selected are:

```
##  [1]  27394 128359  40860   9709 134006  32024 829417  14477  84736 507330
## [11]  48681  39360  20442 129254  50036  71770  19328  95533  32753  12508
```

As we said the sampling is random but it can be seen that the sizes of the cities selected are quite big, if we run the code different time is rare to get a city with less than 10000 inhabitants. In fact, we almost always get at least a couple cities from the Stockholm-Göteborg-Malmö areas.

Let us plot the population histogram of the chosen cities and compare that to the population histogram of all the swedish cities/municipalities.





We see that the histograms have almost the same shape in both cases. Thus, the sizes of the cities/municipalities chosen by the sampler are representative of the sizes of swedish cities.

Assignment 2

- 1 The double exponential distribution is:

$$DE(\mu, \alpha) = \frac{\alpha}{2} exp(-\alpha|x-\mu|)$$

We want to generate a double exponential distribution DE(0,1) from U(0,1) using inverse CDF method.

It is known that the CDF of the double exponential distribution is:

$$\begin{cases} \frac{1}{2} exp(\alpha(x-\mu)) & \text{if } x < \mu \\ 1 - \frac{1}{2} exp(-\alpha(x-\mu)) & \text{if } x \geq \mu \end{cases}$$

In this case since we want a DE(0,1) it becomes:

$$\begin{cases} \frac{1}{2} exp(x) & \text{if } x < 0 \\ 1 - \frac{1}{2} exp(-x) & \text{if } x \geq 0 \end{cases}$$

To find F_X^{-1} we have to solve for x these two equations:

$$\begin{cases} U = \frac{1}{2} exp(x) & \text{if } x < 0 \\ U = 1 - \frac{1}{2} exp(-x) & \text{if } x \geq 0 \end{cases}$$

Solving for the first one:

$$U = \frac{1}{2} exp(x) \text{ implies } 2U = exp(x) \text{ implies } x = \log 2U$$

and $x < 0$ implies $\log 2U < 0$ implies $U < \frac{1}{2}$

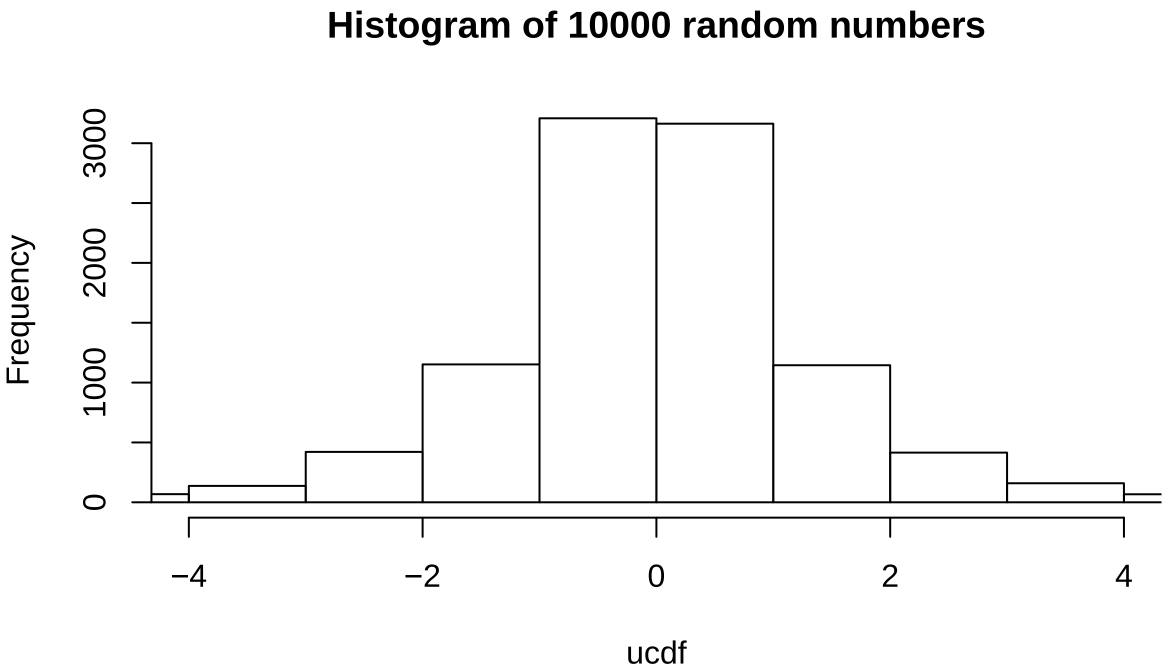
Solving for the second one:

$$U = 1 - \frac{1}{2} exp(-x) \text{ implies } -2(U-1) = exp(-x) \text{ implies } x = -\log(2-2U)$$

and $x \geq 0$ implies $\log(2-2U) \geq 0$ implies $U \geq \frac{1}{2}$

Using this steps we generated our code that can be found in the Code part.

With the code we generate 10000 random numbers from this distribution and we plot the histogram:

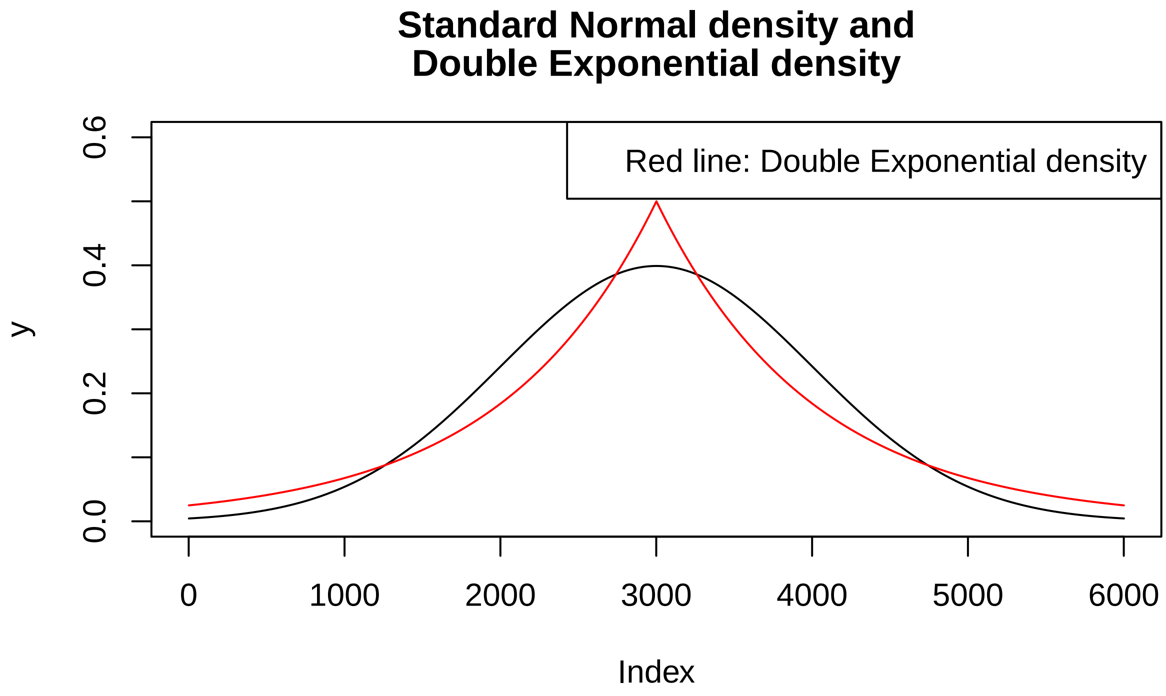


The result looks reasonable since it can be seen that the histogram has the same pattern as the double exponential distribution with parameter $\mu = 0$ and $\alpha = 1$.

- 2

Now, we use Acceptance/rejection method with DE(0,1) as majorizing density to generate target density N(0,1) variables.

The plot shows the two distributions together:



In order to find the costant c we look for the maximum difference where the Normal distribution is greater than the Double Exponential distribution. We find this value to be:

```
## [1] 1.283753
```

The next step is to generate random numbers U from an Uniform distribution U(0,1) and see if the following statement is true:

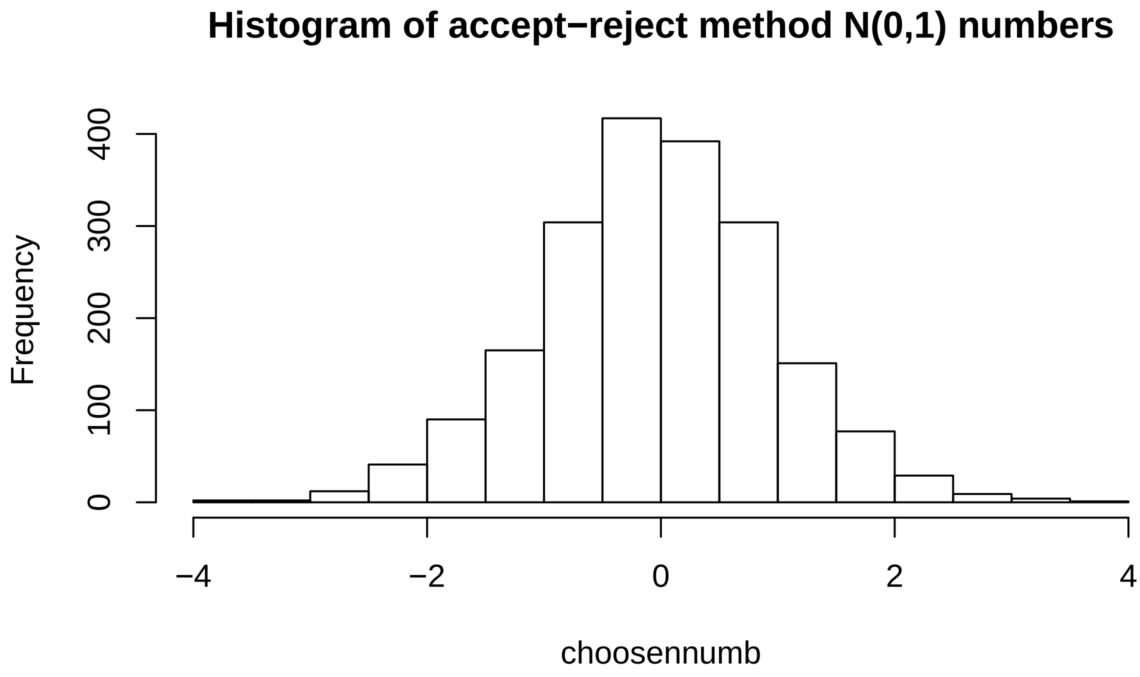
$U \leq \frac{f_X(Y)}{cf_Y(Y)}$

where:

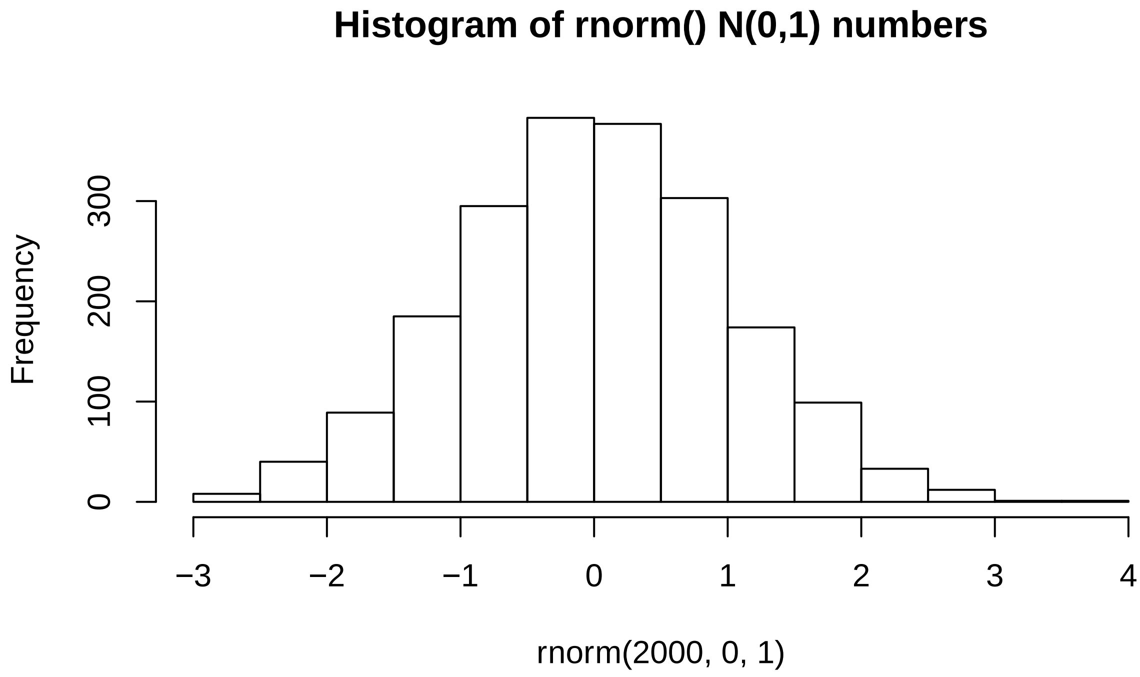
- c is the value found before
- Y is generated from the distribution with density DE(0,1)
- f_X is the N(0,1) density
- f_Y is the DE(0,1) density

The code use these assumptions and can be found in the Code part.

Using the code we generate 2000 cases where the statement above is true. The Y so chosen will be N(0,1) distributed. We plot the histogram of these numbers.



We, than, generate 2000 numbers from N(0,1) using standard `rnorm` procedure and plot the histogram:



The rejection rate R in the acceptance/rejection procedure is:

```
## [1] 0.3015
```

And the expected rejection rate ER is:

```
## [1] 0.2210342
```

It can be seen that the two values are quite similar, but the theoretical has a tendency to be smaller.

Group contributions

We both wrote pieces of this Lab report. The Assignment 1 code is from Thomas lab report, while the Assignment 2 code is from Andreas Lab report. The figures and equations mostly come from Andrea. We also discussed the results and their interpretations. Thomas helped fill the gaps in Swedish geography knowledge required in assignment 1.

Appendix

R code

```
## Assignment 1
library(XLConnect)
wb = loadWorkbook(paste0(getwd(),"/population.xls"))
data = readWorksheet(wb, sheet = "Table", header = TRUE)
data <- data[-which(nchar(data$Statistics.Sweden) == 2),]
data <- data[-(1:4),c(2,4)]
data[,2] <- as.numeric(data[,2])

propcitypicker <- function(cities){
  citypopulations <- data[match(cities,data[,1]),2]
  totalcitypop <- sum(citypopulations)
  binborders <- cumsum(citypopulations / totalcitypop)
  roll <- runif(1)
  if( roll < binborders[1]){
    return(cities[1])
  }
  for(i in 1:(length(cities)-1)){
    if( binborders[i] < roll && roll <= binborders[i+1]){
      return(cities[i+1])
    }
  }
}

cities <- data[,1]
listofcities <-c()
counter <- 1
while(counter <= 20){
  chosencity <- propcitypicker(cities)
  cities <- cities[-match(chosencity,cities)]
```

```r
  listofcities <- c(listofcities,chosencity)
  counter <- counter + 1
}

listofcitysizes <- data[match(listofcities,data[,1]),2]
listofcities
listofcitysizes
hist(listofcitysizes,breaks = 50,main="Histogram of Swedish city populations,chosen cities",
     xlab="Population",ylab = "No. of cities")
hist(data[,2],breaks=100,main="Histogram of Swedish city populations, all cities",
     xlab="Population",ylab = "No. of cities")
u <- runif(10000, 0, 1)

invfunc <- function(u){
  res <- c()
  for(i in 1:(length(u))){
if(u[i] < 1/2){
  invcdf <- log(2*u[i])
  res <- c(res, invcdf)
}else if(u[i] >= 1/2){
  invcdf <- -log(2 - 2*u[i])
  res <- c(res, invcdf)
}
  }
return(res)
}

ucdf <- invfunc(u)

hist(ucdf, xlim =c(-4,4), main = "Histogram of 10000 random numbers")
x <- seq(from=-3, to=3,by=0.001)

#Laplace distribution
doubleexp <- function(x){
  result <- 1/2 * exp(-abs(x))
  return(result)
}

dedistr <- doubleexp(x)
ndistr <- dnorm(x, 0, 1)

#plot of Normal and Laplace together
plot(dnorm(x, 0, 1), type="l", ylim = c(0 , 0.6), ylab = "y",
     main = c("Standard Normal density and", "Double Exponential density"))
lines(doubleexp(x), col = "red")
legend("topright", legend = c("Red line: Double Exponential density"))

#find the c value
diff <- ndistr - dedistr
m <- which.max(diff)
c <- ndistr[m]/dedistr[m]
c
#acceptance/rejection
```

```
newu <- runif(5000, 0, 1)

newucdf <- doubleexp(ucdf)
norm <- dnorm(ucdf, 0, 1)

rate <- norm / (newucdf * c)

choosennumb <- c()
i <- 1
while(length(choosennumb) < 2000 ){
  if(newu[i] <= rate[i]){
   choosennumb <- c(choosennumb, ucdf[i] )
  }
  i <- i + 1
}

hist(choosennumb, main = "Histogram of accept-reject method N(0,1) numbers")
hist(rnorm(2000, 0, 1), main = "Histogram of rnorm() N(0,1) numbers")
#rejection rate R
rejrate <- (i - 2000) / 2000
rejrate
#expected rejection
ER <- 1 - 1 / c
ER
## NA
```